

Appendix- Implementation details

AnyBURL: AnyBURL requires as input a knowledge graph, in the form of tab separated values (tsv) files. Thus, the full Biomedical Literature Knowledge Graph had to be translated into a tsv format file. The groundtruth triples have been divided into ten folds through ten different files. For each repetition, the nine folds have been merged with the main knowledge graph tsv file, and one has been kept as testset. Using the prediction scores calculated by the AnyBURL algorithm, the source Java code of the tool has been edited in order to calculate the Precision, Recall and F1-Score metrics for each fold.

SemaTyP: We have re-implemented SemaTyP in Java, in order to make use of the Java API to the Neo4j database holding the KG. The Java SemaTyP implementation collects all DTD paths relating drugs with targets, and ignores article nodes, MENTIONED_IN relations, as well as triples retrieved only from a single article.

BLGPA: The path collection and SE+PR feature extraction modules have been also implemented in Java exploiting the Java API to the Neo4j database. The random forest classifier has been built in python using the scikit-learn¹ library.

Graph Embeddings: We have used the PyKEEN library² and PyTorch³ to produce the TransE, DisMult, HoLE and RESCAL graph embeddings.

RGCN: For implementing the RGCN classifier, the KG had to be extracted in a tsv file, in order to build the Graph Convolution Network out of it. For this purpose, we have employed the RGCNEncoder of the PyTorch Geometric library⁴.

¹ <https://scikit-learn.org/stable/modules/ensemble.html>

² Ali, M., Berrendorf, M., Hoyt, C.T., Vermue, L., Sharifzadeh, S., Tresp, V., Lehmann, J.: PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research* 22(82), 1–6 (2021)

³ <https://pytorch.org/>

⁴ <https://pytorch-geometric.readthedocs.io/en/latest/>